



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/785,625	02/16/2001	Christine Michelle Barnes	343355600026	4951

7590

12/02/2004

John V. Biernacki
Jones Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER

KANG, INSUN

ART UNIT	PAPER NUMBER
----------	--------------

2124

DATE MAILED: 12/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/785,625

Applicant(s)

BARNES ET AL.

Examiner

Insun Kang

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 July 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-36 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-36 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filed 7/26/2004.
2. As per applicant's request, claims 1, 7, 8, 11, 15, 17, 26, and 27 have been amended and claims 32-36 have been added. Claims 1-36 are pending in the application.

Specification

3. The objection to the specification has been withdrawn due to the amendment to the specification.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
5. Claims 1-25, 27, and 28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Per claim 1, it is unclear as to which object in line 6 it is referring. It is interpreted as "the objects." Accordingly, "its private object state" is interpreted as "their private object states."

Per claim 8, it is unclear as to which public state data in line 9 it is referring. It is interpreted as "the public state data." "Private state data" in line 11 is interpreted as "the private state data."

Per claim 27, it is unclear as to which public state data in line 8 it is referring. It is interpreted as "the public state data." "Private state data" in line 10 is interpreted as "the private state data."

Per claim 11, "private object state data" in the last line is interpreted as "the first and second private object state data."

The term "substantially" in claim 7 is a relative term, which renders the claim indefinite. The term "substantially" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

As per claims 2-7, 9-25, and 28, these claims are rejected for dependency on the above rejected parent claims.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-36 are rejected under 35 U.S.C. 102(b) as being anticipated by Johnson (XML JavaBeans series, Part 1-3, published 2-7/1999, JavaWorld).

Per claim 26:

Johnson discloses:

Art Unit: 2124

- generating a node tree whose nodes store the public and private object state data ("an XML document in a program can be represented as a tree of ... (DOM) nodes... flatten that tree of DOM tree and write it to a text file," pg 5-6 in Part 2; Simpler XML in Part 3; "call the method getAsDOM(), which builds a DOM document tree based on the properties of the JavaBean," pg 7 par 4-5 in Part 2; see pg 6 par 4-5 in Part1; pg 9 paragraph 4-5 in Part 3).
- Wherein an object is queried with respect to its private object state in order to determine the private object state data (pg 5-6 in Part 2; pg 7 par 4-5 in Part 2; see pg 6 par 4-5 in Part1; pg 9 paragraph 4-5 in Part 3)
- processing the nodes of the node tree to generate nodes in an XML tree, wherein the nodes in the XML tree correspond to an XML tag structure ("convert a JavaBean to a tree, and then convert the tree to text, which is then written to a file... The XML corresponding to the DOM tree," pg 5-6 in Part 2; see Figure 3 in Part 2; pg 8 in Part 1)
- generating XML tags based upon the nodes in the XML tree, wherein the XML tags are structured so as to persist the public and private object state data ("XML as a persistence format for JavaBeans components," pg 2 in Part 2; "XMLBeanReader gives... the ability to take a "flat" representation of an object structure, that is, a JavaBean and its properties represented as a text stream that just happens to be XML," pg 3 par. 3 in Part 2; pg 5-6 in Part 2; see pg 6 par 4-5 in Part1; see Figure 4).

Per claim 27:

Art Unit: 2124

The rejection of claim 26 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the public and private object state data
("readXMLBean ... creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3)
- instantiating objects based upon the recovered public and private object state data ("identifies the first element of the document read, and calls instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)
- using the instantiated objects within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3; "controlling XMLBeans properties by integrating XMLBeans with the core java.beans package," pg 2 par 1 in Part 3).
- public state data is data that can be retrieved via public method calls or public fields; private state data is an object's internal state data (pg 7 in Part 1) as claimed.

Per claim 28:

The rejection of claim 27 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the design time object state data
("readXMLBean ... creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3);

Art Unit: 2124

- instantiating objects based upon the recovered design time object state data ("identities the first element of the document read, and calls instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)
- using the instantiated objects within the object development environment such that the recovered design time object state data is used only within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3).

Per claim 29:

The rejection of claim 26 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the public and private object state data ("readXMLBean ...creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3)
- generating source code based upon the recovered public and private object state data (XMLBean can transform an XML document...into a running JavaBean," pag 1 par 2 in Part 3)
- using the generated source code to perform a computer operation (XMLBean can transform an XML document...into a running JavaBean," pag 1 par 2 in Part 3)

Per claim 30:

Art Unit: 2124

The rejection of claim 26 is incorporated, and further, Johnson teaches that the public and private object state data comprise state data from JavaBeans ("an XML file that specifies the values for a JavaBean's properties," pg 7 in Part 1).

Per claim 31:

The rejection of claim 26 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the public and private object state data
("readXMLBean ...creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3)
- instantiating objects in an order based upon the stored state restoration order
("allowing specification of a setAsDOM() method to match any getAsDOM() method....XMLBeanReader uses setAsDOM() to initialize a property value in a bean-object instance from the structure of a DOM subtree," pg 5 par 1-3 in Part 3; pg 9 par 3-7 in Part 1), wherein the instantiating of the object recovers the public and private object state data ("identities the first element of the document read, and calls instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)
- using the instantiated objects within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3).

Per claim 1:

Johnson discloses:

Art Unit: 2124

- determining the private object state data of objects used within the object development environment("XML file that specifies the values for a JavaBeans' properties," pg 7 paragraphs 2-8; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- an object is queried with respect to its private object state in order to determine the private object state data (pg 5-6 in Part 2; pg 7 par 4-5 in Part 2; see pg 6 par 4-5 in Part1; pg 9 paragraph 4-5 in Part 3)
- storing the determined private object state data in a computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2)
- wherein the computer-readable file is in a human-understandable format ("making JavaBeans mobile and interoperable by representing them as XML documents," pg 1 summary;
- restoring the private object state data by processing the computer-readable file ("converts XML files to JavaBeans running in memory," pg 1 summary)

Per claim 2:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the human-understandable format is a text-based format ("Beans as XML documents," pg 7 paragraph 2)

Per claim 3:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

Art Unit: 2124

- after storing the determined private object state data in the computer-readable file, modifying the private object state data within the computer-readable file without using the object development environment ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)
- restoring the stored private object state data by processing the computer-readable file, wherein the restored private object state data contains the modifications to the private object state data ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; "We simply gave the XML parser the name of the XML file, and it returned the entire data structure that the XML file represented," pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)

Per claim 4:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- after storing the determined private object state data in the computer-readable file, directly editing the computer-readable file in order to modify the private object state data within the computer-readable file ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)
- restoring the stored private object state data by processing the computer-readable file, wherein the restored private object state data contains the modifications to the private object state data ("hook functions that allow a

Art Unit: 2124

programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

Per claim 5:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the modifications are to correct errors in object structure without using the object development environment (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section).

Per claim 6:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- an object class that specifies structure of the private object state data has been modified(pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

- said modification occurring after the private object state data has been stored in the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5,

Customization section).

- restoring the private object state data from the computer-readable file even though the object class has been modified (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 7:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

Art Unit: 2124

- an object class that is included in the object development environment and that specifies structure of the private object state data has been modified (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)
- said modification occurring after the private object state data has been stored in the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)
- restoring back into the object development environment substantially the private object state data from the computer-readable file despite the structures differing between the modified class and the private object state data (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 8:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- determining public and the private object state data of the objects used within the object development environment ("XML file that specifies the values for a JavaBeans' properties," pg 7 paragraphs 2-8)
- storing the determined public and private object state data in the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2)
- restoring the private and public object state data by processing the computer-readable file ("converts XML files to JavaBeans running in memory," pg 1 summary)
- public state data is data that can be retrieved via public method calls or public fields; private state data is an object's internal state data (pg 7 in Part 1) as claimed.

Per claim 9:

The rejection of claim 1 is incorporated, and further, Johnson teaches a Java development environment ("JavaBeans," pg 7 paragraph 2 in Part 1).

Per claim 10:

The rejection of claim 1 is incorporated, and further, Johnson teaches a Java development environment for providing graphical user interfaces ("JavaBeans," pg 7 paragraph 2 in Part 1).

Per claim 11:

The rejection of claim 1 is incorporated, and further, Johnson teaches determining that first private object state data is to be restored before second private object state data... computer-readable file is in a structured format that indicates order in which the private object state data is to be restored ("using XML as a persistence format for Javabeans components" pg 2 paragraph 5-6 in Part2).

Per claim 12:

The rejection of claim 11 is incorporated, and further, Johnson teaches: restoring the private object state data in the order specified by the structured format of the computer-readable file (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 13:

Art Unit: 2124

The rejection of claim 12 is incorporated, and further, Johnson teaches that the structured format is an XML structured format (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 14:

The rejection of claim 13 is incorporated, and further, Johnson teaches that the XML structured format includes nested XML blocks to indicate the order in which the private object state data is to be restored (See Figure 8 in Part 2).

Per claim 15:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the computer-readable file is in a structured format that contains private and public object state data ("using XML as a persistence format for JavaBeans components" pg 2 paragraph 5-6 in Part2).

Per claim 16:

The rejection of claim 1 is incorporated, and further, Johnson teaches restoring into a different type of object development environment the private object state data from the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 17:

The rejection of claim 16 is incorporated, and further, Johnson teaches an XML structure such that the computer-readable file is configured for being imported both into

Art Unit: 2124

the object development environment and the different type of object development environment (pg 2 paragraph 5-6 in Part2; pg 4 "Customization" in Part 3).

Per claim 18:

The rejection of claim 16 is incorporated, and further, Johnson teaches:

creating Java objects based upon the restored private object state data, wherein the Java objects are used within the object development environment (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 19:

The rejection of claim 16 is incorporated, and further, Johnson teaches:

- creating Java objects based upon the restored private object state data (see the section "Creating JavaBeans from XML," pg 9 in Part 1)
- storing the Java objects in an object hash table ("The object being created is introspected to get a list of its properties, and then a hash table is created that indexes property names against their corresponding property descriptors," pg 14 paragraph 5 in Part 3)
- retrieving a frame based upon the Java objects stored in the object hash table (pg 14 paragraph 5 in Part 3)

Per claim 20:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

Art Unit: 2124

- restoring the private object state data by processing the computer-readable file (see the section "Creating JavaBeans from XML," pg 9 in Part 1)
- using the restored private object state data to generate source code (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 21:

The rejection of claim 20 is incorporated, and further, Johnson teaches:

using the restored private object state data to generate a different type of source code (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 22:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- (a) determining whether the private object state data of the objects have been modified from the initial values given to the objects upon the objects' creation (pg 4-5, Customization section; pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)
- (b) storing in the computer-readable file the private object state data that has been determined in step (a) to have been modified ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section in Part 3; pg 2 paragraph 6-7 in Part 3)

Art Unit: 2124

-(c) restoring from the computer-readable file the private object state data that has been determined in step (a) to have been modified (pg 4-5, Customization section; pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)

Per claim 23:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-determining customization hooks associated with the objects used within the object development environment ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

-storing the customization hooks in the computer-readable file ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

-restoring the customization hooks by processing the computer-readable file (pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

Per claim 24:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-determining design time object state data associated with the objects used within the object development environment ("use XML as a serialization format for beans," pg 7 paragraph 2)

Art Unit: 2124

- storing the design time object state data in the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2)
- restoring the design time object state data by processing the computer-readable file, wherein the restored design time object state data is used during design time ("converts XML files to JavaBeans running in memory," pg 1 summary)

Per claim 25:

The rejection of claim 24 is incorporated, and further, Johnson teaches:

- determining run time object state data associated with the objects used within the object development environment ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- storing the run time object state data in the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- restoring the run time object state data by processing the computer-readable file, wherein the restored run time object state data is used during run time (pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3).

Per claim 32, it is the computer-implemented apparatus version of claim 1, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 1 above.

Art Unit: 2124

Per claim 33:

The rejection of claim 32 is incorporated, and further, Johnson teaches:

Customizing the serialization of an object without requiring a change to the object ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3) as claimed.

Per claim 34:

The rejection of claim 33 is incorporated, and further, Johnson teaches:

-providing custom hooks for writing out state information of an object ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3) as claimed.

Per claim 35:

The rejection of claim 33 is incorporated, and further, Johnson teaches:

BeanStateInfo object ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3) as claimed.

Per claim 36:

The rejection of claim 32 is incorporated, and further, Johnson teaches:

-serializing objects such that only properties that have not changed from their respective default values are written to the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3) as claimed.

Response to Arguments

8. Applicant's arguments filed 7/26/2004 have been fully considered but they are not persuasive.

Per claim 1:

The Applicant states:

As recited in claim 1, the private state data is obtained by querying an object with respect to its private object state. In this way, the method of claim 1 can allow for storing both private data as well as public data for later restoring. In contrast, the Johnson reference discusses saving public state information and not private state data ...Public state data is data that can be retrieved via such methods as public method calls or public fields. Private state data is an object's internal state data which is not accessible via the object's public interface. Since many objects contain internal states that should be captured in order to properly restore them later (such as runtime), the method of claim 1 allows for the capturing and restoring of this important state data, whereas Johnson cannot (pages 15-6)

In response to applicant's argument, the examiner demonstrates several reasons why the claim is not in condition for allowance:

In response to applicant's argument that the reference fails to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., Private state data is an object's internal state data which is not accessible via the object's public interface) are not recited in the rejected claim(s). The claim does not recite the limitations that distinguish the private state data from public data. The applicant defines the public data as a "data that can be retrieved via such methods as public method calls or public fields" while private state data is "an object's internal state

Art Unit: 2124

data which is not accessible via the object's public interface (page 15)." Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). As such, the claims are read with the broadest reasonable interpretation in mind (Note MPEP 2111). The applicant's argument regarding the definition of the private state data during the initial interview was considered. However, the examiner points out that the limitation defining the private state data is still not recited in the instant independent claims.

Further, the applicant states that the private state data is obtained by querying an object with respect to its private object state so that storing both private data as well as public data for later restoring can be allowed. This statement is contradictory with the later statement defining the private state data as an "object's internal state data which is not accessible via the object's public interface". That being said, how can the private state data be queried when the data is not accessible? The claim does not recite as to how this querying is performed. Therefore, the examiner interprets that the querying is performed in a conventional way. Accordingly, it can be concluded that if the private state data can be queried as in the instant invention but simply can be chosen not to be stored for encapsulation, Johnson's invention is capable of storing the private state data along with the public data.

Further, in object-oriented programming, objects encapsulate their private state and run a common task by invoking each other's interface. In Johnson, a bean stores the persistent state for all its public properties. However, with serialization, the internal

state of the bean may also be persisted. As the internal state can be essential for some cases in order to save the object state so that the object can be restored with original or similar characteristics. However, without the control of human readability, the private state data, which may contain a sensitive data, can be exposed in public eyes if the serialization formatter is human understandable. The claim recites that the formatter is "human-understandable" and does not recite any selection of the private data to serialize. Therefore, this exposure of the private data directly violates encapsulation and leads to harm the public interest where the sensitive data can be read by anyone (or even modified). The claim does not recite that the serialization is performed without breaking encapsulation nor recite control of the degree of the human-readability in order to hide the private data that should not be exposed. Therefore, the claim is not directed for the public interest and accordingly considered not in condition for allowance for several reasons addressed above and the rejection of claim 1 is maintained in view of the broadest reasonable interpretation.

Per claims 26-31:

The applicant states that Johnson does not disclose the limitations of claim 26 for the reasons set forth in connection with claim 1. As shown above, the rejection of claim 1 by Johnson was maintained, and accordingly, the rejection of claim 26 is also maintained. The applicant states that claims 27-31 are allowable as being dependent on allowable base claim. As has been shown above, the rejections of the independent

Art Unit: 2124

claims 1 and 26 by Johnson are maintained, the argument that claims 27-31 are allowable as being dependent on an allowable base claim is considered moot.

Accordingly, the rejections of claims 27-31 are considered proper and maintained.

Per claim 11:

The applicant argues that Johnson does not disclose that a restoration order is determined such that first private object state data is to be restored before second private object state data because of an interdependency between the first private object state data and the second private object state data (page 16).

In response, the examiner points out that Johnson discloses a XML formatter. XML document is human readable, and therefore the object structure can be easily manipulated. Therefore, the XML formatter in Johnson allows the storing of object state data with a specific restoration order. Accordingly, the rejection of claim 11 is maintained.

Per claim 24:

The applicant argues that:

Determining design time object state data associated with the objects used within the object development environment so that the design time object state data can be later restored for use during design time. In contrast, the Johnson reference does not disclose an object writing out two different representations of itself such that it is restored differently in design-time and run-time contexts (page 17).

In response, Johnson discloses the XML formatter for serialization. As has been addressed above, XML is human-readable and therefore the object structure can be easily manipulated. Therefore, the XML formatter in Johnson allows the storing or restoring of object state data in a specific way. Accordingly, the rejection of claim 24 is maintained.

Conclusion

9. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 571-272-3724. The examiner can normally be reached on M-F 9:30-6.

Art Unit: 2124

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 571-272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

I. Kang
11/24/2004

Kakali Chaki

**KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100**